

N 9 2 - 2 1 8 9 1

NEURAL NETWORK CLASSIFICATION OF "QUESTIONABLE" EGRET EVENTS

C. A. Meetre¹ and J. P. Norris²
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

ABSTRACT

High-energy gamma rays (> 20 MeV) pair producing in the spark chamber of EGRET give rise to a characteristic but highly variable 3-D locus of spark sites, which must be processed to decide whether the event is to be included in the database. A significant fraction ($\sim 15\%$: 10^4 events / day) of the candidate events cannot be categorized (accept / reject) by an automated rule-based procedure, are therefore tagged, and must be examined and classified manually by a team of expert analysts. We describe a feed-forward, back-propagation neural network approach to the classification of the questionable events. The algorithm computes a set of coefficients using representative exemplars drawn from the preclassified set of questionable events. These coefficients map a given input event into a decision vector that, ideally, describes the correct disposition of the event. The net's accuracy is then tested using a different subset of preclassified events. Preliminary results demonstrate the net's ability to correctly classify a large proportion of the events for some categories of questionables. Current work includes the use of much larger training sets to improve the accuracy of the net.

INTRODUCTION

The Energetic Gamma Ray Experiment Telescope (EGRET) on the Compton Gamma Ray Observatory is sensitive to high-energy gamma rays in the regime 20 MeV to 30 GeV. Two vertically stacked spark chambers comprise the primary detector. A photon is detected when it interacts in one of several thin tantalum plates in the upper spark chamber, producing trajectories which, propagating downward, are revealed by sparks where the leptons interact in a series of decks (interleaved with the tantalum plates). Each deck contains two orthogonal, horizontal grids of closely spaced wires with a high voltage differential. A magnetic core readout system is used to determine the spark positions. An anti-coincidence scintillator dome surrounding the upper portion of the telescope and a directional time-of-flight coincidence system in the lower portion discriminate against cosmic rays and Earth albedo. Details of the EGRET instrument and its scientific objectives are described in Kanbach *et al.* (1989) and Fichtel *et al.* (1989), respectively.

A complex ruled-based program, Search and Analysis of Gamma Ray Events (SAGE), definitively processes a large proportion of candidate events. SAGE eliminates virtually all cosmic-ray events, and categorizes apparent gamma-ray events as either 'acceptable' or 'questionable'. About 15% of all events are questionables. These are examined visually and classified by a team of expert analysts. For both SAGE-accepted events and questionables which are accepted by analysts, SAGE characterizes the photon arrival direction and energy using the spark positions and the energy deposition measured by a large NaI(TP) scintillation crystal below the lower plastic scintillator.

¹ STX, Code 690.2 ; ² Code 668

Although the manual processing task for questionable events is quite tractable, it is labor intensive, and – given the large data volumes – expensive and time consuming. In addition, human analysts are prone to variable performance. Rule-based and expert system approaches have been thoroughly investigated over the years and have not yielded satisfactory results on the set of questionables.

We have explored neural networks for processing some categories of EGRET questionable events. The fact that a trained human recognizes events as 'accepts' or 'rejects' implies that a neural network trained on a set of events categorized by analysts could perform the same feat. Because the EGRET data set is so large, the up-front effort involved in developing the network can be amortized to yield a very cost-effective solution. A network can process in minutes what an analyst can do in several months.

NEURAL NETWORKS

Neural networks are an artificial intelligence technique that applies an algorithm modeled on the way the human brain learns to discriminate among patterns perceived by the senses. In general, the brain receives sensory information (its input data) and produces an estimate of the classification of the pattern (its decision vector). This estimate is corrected by some training agent, by comparison with the ideal answer (its target vector), and the estimate is reinforced if correct. Over many repetitions the brain comes to recognize, for example, a picture of a horse as a horse, and can then extend this to other pictures of horses, provided the original sample of pictures showed a wide enough variety of horses and orientations.

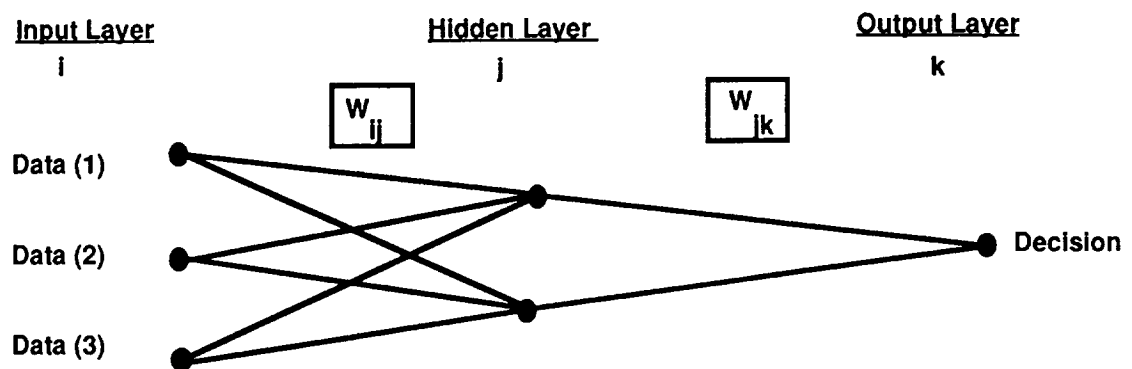
It is clear that the human brain, when recognizing patterns, does not usually use a rule-based system. The brain's cycle time is relatively slow, and rule-based 'if-then-else' systems are inherently serial in nature. The number of decisions necessary to apply an 'if-then-else' algorithm to visual pattern recognition would vastly overrun the actual 0.5-s response time of the brain. Thus, pattern recognition is accomplished by the brain in a massively parallel manner using many processors, many connections, and few cycles. Recognition knowledge appears to be embedded into the brain by the amplification or attenuation of the pathways over which the signals travel between successive neurons. Not surprisingly, this type of learning behavior can be simulated by an electronic circuit. And, more importantly for our purposes, it can be simulated by a computer.

This simulation requires the algorithm shown graphically in Figure 1. Vectors are used to represent the input to and output from neurons. Neurons are simulated using the sigmoid function, $f(x) = 1/(1 + \exp[-x])$, where $f(x)$ is the output from a neuron whose collective input is x . This function models closely the response and thresholding activity of a neuron. A weight, or coefficient, is associated with each connection among the neurons and it is this coefficient that amplifies or attenuates the information associated with the connection. These coefficients 'store' the knowledge of the net in a non-explicit form and are changed by small increments to allow the net to converge to a correct outcome.

A network is trained using an appropriately sized set of exemplars which previously have been classified. The size of this set will depend on the complexity of the pattern to be classified. The selection of patterns must cover the possible range of input data. Once the training is completed, the network can be used to classify new patterns drawn from the range of training patterns.

The selection of exemplars is a major part of the up-front effort required to use a neural network and means that networks are practical in application only where considerable amounts of data are to be processed. In addition, it should be noted that there is no physical information embedded in

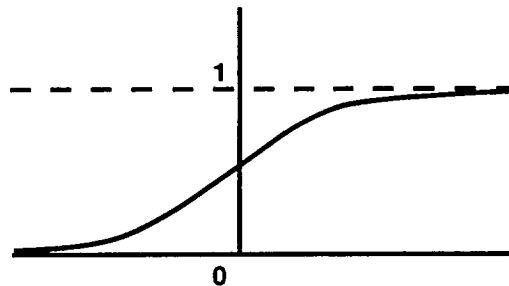
A Simple Three-Layer Neural Network



The Activation Function

$$x_j = \sum \text{data}_i \cdot w_{i,j}$$

$$f(x_j) = \frac{1}{1 + e^{-x_j}}$$



This function acts as a processor, and produces an output between 0 and 1.

Fig. 1– Connectivity of a small neural network with one layer of hidden nodes. The activation function closely models the threshold response of a biological neuron.

the coefficients. The network is simply a classifier and gives no analytical information about physical processes.

Because the function used is general, although variable in the number of terms it contains, the software used to train the network is reusable for any problem. It is surprisingly uncomplicated, and, in this application, was implemented in Fortran on a VAX 3100. In a rule-based system, the program must be written uniquely for each problem. Even with expert systems, an explicit knowledge base must be developed for each problem, though the software that handles the knowledge is reusable.

A neural network is a non-linear least-squares fit to a very general function (h) in which the independent variable is an array of experimental data (X), and the dependent variable is an array of characteristics (D) often called the decision vector – the classification. During training, the decision vector is compared to a target vector (T) which contains the correct classification for the event. The error vector thus generated (E) can be used to calculate the relative contribution to this outcome of all the coefficients (a vector W) in the network, and to effect a small change ΔW in a direction that will minimize the error. The relationships are, therefore:

$$h(X) = D$$

$$\text{or } h(x_1, x_2, \dots, x_n) = (d_1, d_2, \dots, d_n) ,$$

$$E = |T - D| \text{ and } \Delta W = g(E) .$$

A value of $d_i \equiv 1$ indicates the presence of characteristic i . Conversely, a value of $d_i \equiv 0$ indicates its absence. The order and number of characteristics is fixed in the formulation of the problem and remains constant. Note that the sigmoid function is applied many times in $h(X)$.

The fitting process determines an optimal value for the coefficients of the function $h(X)$. After the coefficients are determined and inserted in the function, it can be evaluated for a given array of data. The value of the output array is an estimate of the characteristics of the data.

PREPROCESSING EVENTS AND TRAINING THE NETWORKS

The added advantage of human analysts over an automated, rule-based system is the capability of visual pattern recognition. An expert analyst can immediately see which sparks are extraneous to the expected gamma-ray signature, whereas rule-based decision-making sometimes misses the obvious. In reviewing questionable events, analysts continue to apply the rules that SAGE demands for event acceptance. The analyst's role reduces both pattern recognition and respect for SAGE rules to one final decision: accept or reject. We utilize the analysts' decisions on questionable events as input to train the network. Clearly, the network can perform, on average, only as well as the analysts.

The most frequently encountered signature of pair production in the EGRET is an upsidedown 'y' (miniscule): When the gamma ray converts to a pair, both leptons continue practically undeviated until one of them undergoes a major Coulomb collision in a tantalum plate. The two paths diverge at that point (the vertex), forming the two arms of the 'y', the undeviated path above the vertex forming the leg of the 'y' (see Figure 2). Additional scatterings may occur, and therefore the paths near the vertex are required to have straight segments of minimum length so that the event may be characterized. If neither lepton scatters, the signature is a single, undeviated track.

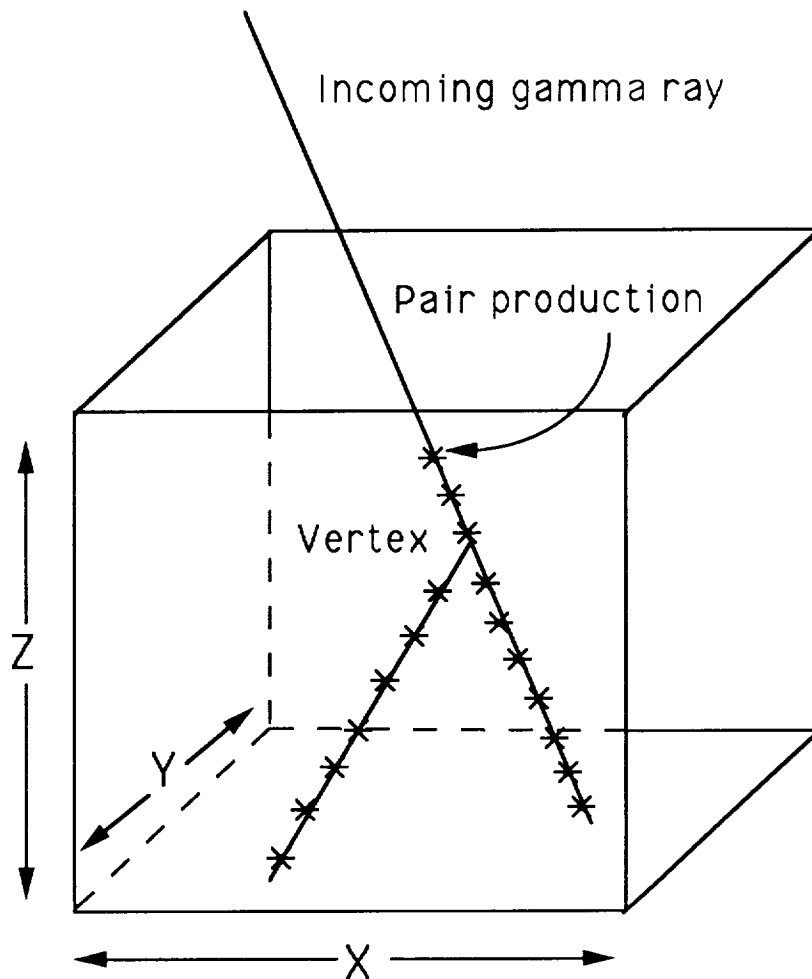


Fig. 2- Schematic of characteristic signature of pair production in the EGRET spark chamber. One lepton usually undergoes a major Coulomb scattering, resulting in a characteristic locus of sparks resembling an upsidedown 'y'.

EGRET data is best represented visually to an analyst as a sparse, large, 3-D matrix shown graphically in two views, X and Y (the XZ and YZ planes). But a more compact representation is necessary for a feasible neural network. Data representation is complicated by the fact that a given deck may have any number of sparks, including none. Furthermore, a spark in the X view cannot unambiguously be associated with a spark in the Y view, unless the spark is the only one on a deck.

Spark information was utilized from 34 decks (stacked in the Z direction) of the upper and lower spark chambers. In both the X and Y directions, there are 992 cores; cores which are 'on' (e.g., 'fired') indicate the paths of the leptons. Usually several adjacent cores fire on each deck when a lepton passes. A string of adjacent fired cores on a deck is called a spark. Two strings of fired cores with an interjacent gap of up to three unfired cores were treated as one spark. The central core of such a string of cores and gaps was then adopted as the X or Y coordinate of the spark.

In the majority of questionable events the recorded pattern has missing sparks and/or extraneous sparks. This kind of 'fuzziness' is difficult for rule-based systems to handle, whereas analysts and neural networks can more easily recognize a good event in the presence of flawed information.

Because a parsimonious representation is desirable, and because the network requires a constant number of independent variables, we retain exactly two sparks per deck. Some information is therefore lost when three or more sparks are present. The rules for retention of sparks per deck are: (1) If no spark is present, zero-fill both positions. (2) If only one spark is present, duplicate its position. (3) If there are two sparks, fill the respective positions with the spark coordinates. (4) If there are more than two sparks, retain the two sparks closest to the {1 spark, 2 sparks, or center} of the preceding (higher) deck. Since one spark per deck is usually present after pair production and before scattering, the duplicated portion of the trajectory may be thought of as representing the paths of the two unscattered leptons.

A single network was trained to recognize the existence of the following states, each of which corresponds to a SAGE rejection criterion: (1) an event appears to enter through a spark chamber wall rather than through the instrument aperture; (2) an event has too few sparks, thus the spark pattern is not sufficiently well-defined; (3) the event separation vertex appears to be on different decks in the X and Y views (thus compromising the characterization of event attributes). Hence, in experiments to date a network was trained with four output nodes: {accept, too few sparks, wall, or different decks}. In future experiments, we plan to train one network for each rejection criterion (e.g., only two output targets per network: accept or reject), and to be accepted an event will have to pass each network in succession.

Both the X and Y views are reviewed simultaneously by the analysts. However, the flaw in a pattern which leads SAGE to question an event may be present in only one view, in which case only the flawed view is used in the training set. Including the apparently acceptable view would introduce contradictory information and tend to influence the net in the wrong direction. For example, a wall event is usually apparent in only one view. In the recognition phase, both views would be required to pass muster (serially) for the event to be accepted by the network. In order to keep the number of input nodes small, we present the two views separately.

In addition to the rejection criteria, three classes of events are distinguished, and each must be processed with a different network. Events which exhibit the characteristic 'y' pattern in both views are called 'doubles'. Events for which SAGE cannot characterize both arms of the 'y' in both views are referred to as 'single-doubles' – apparently single in one view, double in the other. For very high energy gamma rays no separation may occur, or the separation angle may be negligible. These events are 'singles'. For inflight data, about 40% of the questionable events are singles, and the vast majority of these are rejected by analysts. Thus, a solution to the singles problem alone would be very useful. Since very few accepts are found among singles – none in ground-based calibration data, and very few in source-free inflight data – we have not yet been able to train a network to recognize singles. However, this problem seems eminently tractable since it is basically a task of recognizing well-formed straight lines. About 30% of questionables fall in the single-double category. Most of these events are in fact double in both views; however,

Currently, doubles are correctly recognized as accepts in about 60% of the cases; thus about 40% of accepts are falsely rejected, and would still require examination by human analysts. Since only exploratory experiments have been carried out, we expect some improvements as experience is gained in understanding how to present spark pictures to the nets.

FUTURE EXPERIMENTS

We also expect results to improve when larger inflight datasets are used to train the net. From the number of degrees of freedom in the problem, {2-D aperture, angle of incidence, vertex deck, and gamma-ray energy – related to vertex separation angle}, we estimate that it will be necessary to use many thousands each of accept and reject targets in order to train the networks sufficiently well. The distribution of energies for ground-based and inflight data is different. Thus, well-characterized experiments await sets of questionable events from the inflight database that have been processed by human analysts. Only datasets with weak sources or no source in the field-of-view are usable since sources tend to bias the associated incidence angles.

In a source-free field of view, much less than 1% of questionables are acceptable singles. Since questionables account for about 15% of the EGRET database, in order to find ~ 1000 good singles – a minimum requirement to train a network for any given target – we would have to search through a few million events of inflight data. However, the singles problem may very well be soluble by simulating acceptable singles – straight lines – using appropriate numbers of extraneous and missing sparks, as well as small deviations from straightness which occur as a result of electron scattering suffered by the pair. A simulation program for acceptable singles is currently in development.

Several improvements are under consideration. Once the 'double in both views' networks are optimized, it should be straightforward to address the 'single/double view' problem by making appropriate changes in some accept and reject targets. An efficient way to account for left/right symmetry may be to train networks using both an event and its mirror image about the Z-axis. The reject category which involves the separation vertex apparently being on different decks in the two views may require combining the X and Y views. Finally, additional rejection criteria have been identified in the inflight dataset which may be accommodated by adding one reject target.

REFERENCES

- Fichtel, C. E. *et al.*, 1989, Proc. GRO Science Workshop, ed. W. N. Johnson (Greenbelt, MD: NASA/GSFC), p. 3-1.
Kanbach, G. *et al.*, 1989, Proc. GRO Science Workshop, ed. W. N. Johnson (Greenbelt, MD: NASA/GSFC), p. 2-1.

SUGGESTED READING ON NEURAL NETWORKS

- Parallel Distributed Processing, eds. D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, MIT Press, Cambridge, MA, 1986.
Fahlman, S. E., and Hinton, G. E., Connectionist Architectures for Artificial Intelligence, IEEE Computer, 1987.
Lippman, R. P., An Introduction to Computing with Neural Nets, IEEE ASSP Magazine, April 1987, p. 4.